A Volumetric Geometry and Topology Parameterisation for Fluids-based Optimisation

J. Hall[a], T.C.S. Rendall[a,*], C.B. Allen[a], D.J. Poole[a]

[a]*Department of Aerospace Engineering, University of Bristol, BS8 1TR*

Abstract

A new geometry and topology parameterisation method is presented which is based on creating a parameterisation grid of cells and reconstructing surfaces from the fraction of the cell volume defined to be solid, with the volume fractions acting as design variables. This method is able to include topological changes alongside fine-level geometric control, and therefore offers a significant increase in flexibility. In this work, the geometric capabilities of the method are confirmed by successfully constructing a variety of surfaces, using both arbitrary object outlines and aerofoil geometries. The method is then used in a range of optimisation problems covering the design of a coastal defence, increasing fluid damping within an oscillating box by the addition of baffles, and design of a multi-body configuration for minimum drag in supersonic flow. These problems demonstrate the benefits of a parameterisation for fluids modelling that is capable of topological changes and which can be used with global search as well as gradient-based methods.

Keywords: parameterization, optimisation, topology, volume of solid, level set, phase-field, material distribution

*Corresponding author

Email addresses: James.Hall@bristol.ac.uk (J. Hall), Thomas.Rendall@bristol.ac.uk (T.C.S. Rendall), C.B.Allen@bristol.ac.uk (C.B. Allen), d.j.poole@bristol.ac.uk (D.J. Poole)

1. Introduction

In optimisation problems a key challenge is how best to represent trial geometries in a way that maximises coverage of the design space and which promotes good convergence of the optimiser. Methods of representing geometries are often referred to as shape parameterisations as the geometry is represented by a vector of parameters; it is these parameters the optimisation method subsequently adjusts to produce new geometries. This process and the links between the methods are shown in figure 1.

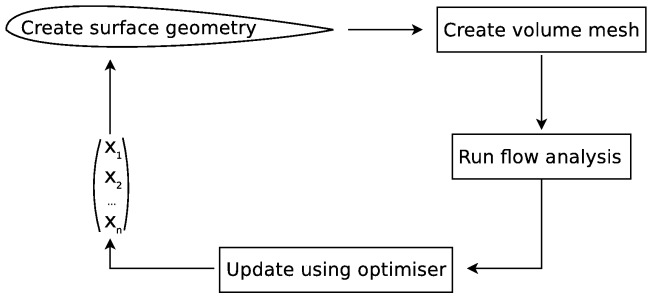


Figure 1: Diagram of interaction between design variables, $\mathbf{x}$, geometry parameterisation scheme, generated shape and an optimisation method

Having the ability to change topology is a desirable but uncommon feature for a parameterisation scheme. Enabling topological change in an optimisation process is desirable because it allows consideration of designs that would not otherwise be accessible [1]. In the case of high lift configurations for aircraft or racing cars, for example, it is not obvious how many lifting elements may be optimal. Equally, the aerospike design used on Trident II missiles [2, 3] would be very difficult to obtain without a parameterisation capable of substantial geometric and/or topological change.

The objective of the work presented is to define a parameterisation scheme that implicitly handles topological change alongside fine-level geometric control. Following a review of existing parameterisation methods in section 1.1 a new

parameterisation method is proposed in section 2. The method is then applied to a set of geometry reconstruction problems in order to explore the accuracy and flexibility of the parameterisation, before testing on optimisation problems including the design of minimum drag configurations in supersonic flow, the design of a coastal defence and design of baffles to increase damping in an oscillating tank.

1.1. Existing Parameterisation Methods

Parameterisation raises two fundamental and interlinked problems; first, how to devise a geometry method that designs shapes suitable for the chosen optimisation, and second, how to have confidence that a sufficient but not excessive number of design variables have been introduced.

A parameterisation is a method of representing a geometry by means of a vector of design variables, and these methods can be broadly split between two groups. The first group seeks to *construct* shapes from empty space, while the second aims to *deform* an already existing geometry. Although distinctions can blur, surface point control, level sets and descriptive function approaches are constructive, while control point techniques are deformative (an exception to this is the way in which NURBS can be used to both construct surfaces from control points, and then also to deform them through motion of those control points). It is clear that a constructive approach may usually be linearised to provide a deformative route providing the initial shape may be encoded, but there is no guarantee a deformative technique will be able to construct a shape from scratch.

Once a parameterisation has been selected, it is possible to consider design space (dimension) reduction (DSR) in an effort to ensure the number of design variables is minimised. The first approach, proposed by Diez *et al.* [4, 5], is to randomly sample a wide range of geometries constructed via the chosen

parameterisation and use a Karhunen-Loève expansion (although other matrix approaches may also be used [6]) to find a series of eigenvectors and values ordered in terms of decreasing importance. From this, a subset of the 'early' eigenvectors may then be used as design variables, and this approach has been applied for marine hull design [4, 5] using free-form deformation. A second approach is to take a representative geometric library and parameterise this with any method through a fitting procedure, before interpolating the calculated parameter values against a chosen, smaller set of variables of interest. For example, an aerofoil library might be parameterised using CST [7], and then the CST parameters interpolated in terms of lift coefficient and thickness to chord ratio, as demonstrated by Sóbester and Powell [8]. It is an important result of these approaches that even if a parameterisation initially introduces too many design variables, these can later be reduced to an acceptable level through DSR. It is therefore not compulsory to address accuracy (ability to reconstruct any shape) and efficiency (ability to reconstruct using a small number of design variables) simultaneously, although clearly a link will always exist.

The goal in this section is to illustrate where an approach handling topological change as part of the parameterisation can fit in to the existing tool box of methods, so the relative merits of existing methods and their most important results shall now be considered.

1.1.1. Surface Points as Design Variables

A natural approach is to consider moving every surface point, so that the design variables are simply the surface point locations in x,y,z. Doing so ensures that the complete design space is retained, as any surface can be represented (in a discrete sense) if all surface points are free to move.

This has been applied extensively in aerodynamic optimisation [9, 10, 11, 12, 13]. The gradient of the objective function must however be smoothed in order

to ensure smooth surfaces are generated, as with such a high number of design variables is it easy to produce noisy surface shapes. It can also be demonstrated on a test example, such as the brachistochrone problem, that in the absence of a smoothing operation surface quality will progressively deteriorate [10].

The benefit of being able to represent a large design space creates difficulties as high dimensional spaces are time consuming to explore, with slow optimiser convergence being common, and it is imperative to use an adjoint route for gradient computation. Using this parameterisation it is also not clear how to create topological changes moving from the starting surface in the absence of any other descriptive mechanism; furthermore, including topological variation in an adjoint frame work would be difficult. Topological derivatives have been defined for general optimisation [14], but not applied with a surface point parameterisation.

1.1.2. Level Sets

Level sets represent the boundary of a geometry as a level set of a function and their use in geometry optimisation is reviewed by van Dijk [15] *et al.* For reasons of convenience the zero level set is usually chosen as this can be detected by a sign change. The use of level sets as a method of representing fronts was first suggested by Osher and Sethian [16] with applications to tracking the behaviour of propagating flame fronts, and this led naturally into Sethian and Wiegmann's [17] work using level sets to parameterise a geometry for optimisation. The level set method operates by evolving the level set function in a time like manner using the Hamilton-Jacobi equation such that at each time step a new geometry is produced as the position of the zero level set changes. The evolution of the level set function is controlled by a velocity term which acts normal to the surface and typically the level set function is initialised as a signed distance function from the initial surface geometry.

Level set based optimisation has been largely applied to steady state structural problems and often requires an adjoint solution. This is especially restrictive when considering the use of an optimisation scheme with a general solver as code specific modifications must be made to solve both the adjoint problem along with the original objective function evaluation.

Sethian and Wiegmann [17] calculated velocity directly from the local stress such that the boundary moves to remove the maximum amount of material subject to constraints; this approach was applied to the optimisation of a cantilever beam. Wang *et al.* [18] found velocities by using the adjoint method to calculate the sensitivity of the objective function to the geometry defined by the level set; this sensitivity was then used to define the level set velocity. Allaire *et al.* [19] present similar results but, as with methods described below in section 1.1.5, used low density material rather than a true absence in regions defined as outside the part. Allaire *et al.* [20] show that level sets evolved by the velocity method are unlikely to add new holes and so create a different topology; this can be improved by the inclusion of the topological derivative [14]. Rather than move a boundary defined by a level set, Wei and Wang [21] define a piecewise constant level set function which has the advantage of easier topology changes through creation of holes.

In addition to the evolution based level set parameterisation, a less common explicit level set approach has been used in optimisation such as that suggested by Kreissl *et al.* [22] where weighted radial basis functions (RBFs) were used to represent the level set function with the weights acting as design variables. Kreissl *et al.* combined this parameterisation with a lattice Boltzmann flow solver to optimise pipe geometry for minimum pressure loss. This method allows standard gradient based solvers to be used with a level set approach but a large number of design variables are needed (on the order of 200) to produce a simple

final geometry and as presented the method cannot be used to produce solid regions in areas that were previously entirely void. Further work by Kreissl and Maute [23] developed a framework based on the extended finite element method (XFEM) which permitted changing topologies to be included in an optimisation based on a Navier-Stokes solver, as XFEM permits discontinuities in shape functions.

1.1.3. Control Point Methods

An intuitive parameterisation is to have a surface constructed by joining a set of control points on that surface [24]. However, unless a large number of points are used the geometry is unlikely to be sufficiently smooth, but conversely a large number of points allows high frequency deformations of the surface. The number of design variables can be reduced by constructing the surface as an interpolation between control points [25] though for linear interpolation this does not solve the smoothness problem and for higher order polynomial interpolation high frequency oscillations may be a problem due to over fitting. Splines can also be used to parameterise surface geometry [26], and B-Splines or Bézier curves [27] are early examples. Directly parameterising the surface in this manner still demands remeshing for each new geometry.

Methods that define the surface of the geometry (constructive) directly are most intuitive but it is also possible to produce a scheme whereby the control variables deform an existing geometry or even a computational mesh produced from the geometry. Free-form deformation (FFD) was first designed for use in computer graphics [28] and describes a relationship between a local coordinate system which defines the positions of mesh nodes, or any other point, and a set of control points arranged in a box around the mesh nodes. Thus any movement of the control points causes movement of the mesh nodes. The position of these

control points can be useful as design variables [29, 30]. A similar method is based upon radial basis function (RBF) interpolation [31]. Here a set of control points are chosen, the position of which act as design variables as shown in figure 2, and a global relationship is built between the position of the control points and the mesh nodes. This allows smooth deformation of the computational mesh and thus the surface geometry in response to the movement of the control points.

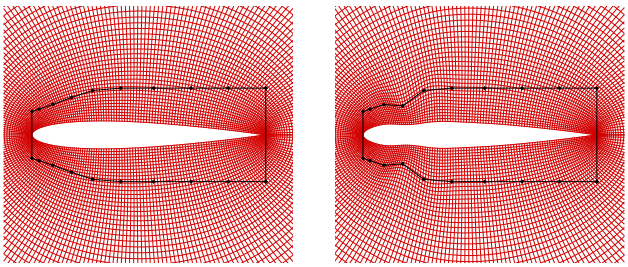


Figure 2: Radial basis function control points undergoing movement to deform an Euler flow mesh [32]

Control point methods are limited in the topology that they can represent. The interpolation based methods are also limited to geometry that resembles the initial geometry, implying that their utility is limited to cases where it is thought that something close to the optimum is already known. A good example of this is the use in aerofoil optimisation; the basic geometry is known from experience but significant improvements to an objective function can be made by small geometry changes.

1.1.4. Descriptive Function Methods

Kulfan and Bussoletti [7] present the class function/shape function transformation (CST) method for representing smooth geometries by the combination of a class function and a number of shape functions. The use of class functions allows several different basic types of body to be represented by modifying the behaviour at the trailing and leading edges, for example aerofoils with sharp or

rounded trailing edges. The shape functions are scaled by the design variables and then control the shape of the rest of the body subject to the constraints applied by the class function. The method has been successfully applied to the problem of inverse design [33] of an aerofoil where the objective is to match a target pressure distribution and to the drag minimisation of a supercritical aerofoil [34]. Although capable of representing a wide variety of geometries, the method cannot represent multiple topologies without direct user intervention.

The Parametric Section (PARSEC) method [35] uses a set of basis functions to build up aerofoil shapes with control provided by design variables that are explicitly linked to physically meaningful parameters of the aerofoil such as leading edge radius of curvature. Additional terms can be added for blunt trailing edges or local surface bumps and the method can be extended to high lift configurations with multiple bodies [35], though additional design variables must be added and the method cannot produce a multiple body geometry without being instructed to do so. This parameterisation method has been applied to the optimisation of a transonic wing for maximum lift to drag ratio by means of a genetic algorithm [36].

The Hicks-Henne [37] method constructs aerofoil surfaces by applying perturbation functions to a base shape and in the initial work the functions were chosen to represent changes in camber or thickness at points along the chord. The design variables are coefficients multiplying the perturbation functions that scale the effect of each function. By parameterising the wing design in this way, Reuther *et al.* [13, 38] optimise the wing design of transonic and supersonic aircraft.

Deformations may also be calculated using a modal decomposition of a geometry library [39], and applied using a control point method [40], or as shape changes directly. When considering aerofoils, public libraries of well over 1000

may be formed, representing a wealth of design experience aimed at the problem in hand. Industrial libraries are likely of similar size but remain proprietary. The modal approach (via singular value decomposition) is arguably the most efficient technique for representing aerofoils, giving the smallest reconstruction error for the smallest number of design variables [41]. Of course, this approach is only suitable if a known library exists, and by its nature cannot handle topology changes.

1.1.5. Volumetric Methods

From the previous review it is clear that there is a significant shortcoming in parameterisations for external aerodynamics not being unable to handle topology change. Volumetric methods have yet to see significant use in shape parameterisation for fluids though they are commonly used to reconstruct surfaces in the volume of fluid (VOF) method often applied for modelling free surfaces [42]. The characteristic of these schemes is that the volume fraction of material is stored in cells in the grid and surface information is then reconstructed from this volume fraction information. In computer graphics, the simple line interface calculation (SLIC) method [43] produces "staircase" surfaces as it seeks to produce an interface in a partially full cell aligned with either grid axis. The piecewise linear interface calculation (PLIC) [44, 45] method is similar in that it uses a linear segments to represent interfaces but this line is not fixed to be orientated to the grid axis but rather the normal of the line is set to the normal of the fluid interface, calculated by a finite difference over adjacent cells. Volume-based methods implicitly capture topology as volume cells can be specified as solid in any configuration.

One, relatively common, volume based method ignores the notion of surfaces altogether and instead adds a variable parameter to the governing equations that controls the extent to which a given point is solid. The physical interpretation

of this parameter varies depending on the problem but in solid mechanics, for example, it might be the density of the material in a computational cell. Cells with low density are essentially void as they can carry little of an applied load, in this way the geometry and topology of the design can be controlled by setting the density in some regions to zero to denote areas without material. The design variables are then equal in number to the computational cells in the domain. For example Borrvall and Petersson [46] performed optimisation on a Stokes flow problem by introducing a parameter which controls the porosity of a region (analogous to the density variable used for structural problems). The extreme values of this parameter correspond to either an impermeable solid or a completely open region and the parameter is biased such that the final solution is either fully solid or fully void at a point. Techniques of this type have also been applied to fluids optimisation using lattice-Boltzmann methods [47, 48, 22, 49] and to solid mechanics shape optimisation by letting the design variable parameter control the density of the material [50, 51]. However, these methods are not suitable for use with black box solvers due to introduction of extra terms in the solver for the variable material properties. Additionally, the number of design variables used is often on the order of the number of cells in the computational grid which makes finding the gradient of the objective function with respect to the design variables computationally impractical with a finite difference based method. Instead, the sensitivity is often calculated via the adjoint method which allows the sensitivity of the objective function to the design variables to be calculated by solving one numerical problem with a cost independent of the number of design variables. Implementing the adjoint method, however, requires detailed control of the solvers.

There is therefore a strong argument for volume based parametrisation schemes which are completely independent of the flow solver used, and which

can also produce smooth surfaces suitable for use in conventional body-fitted finite-volume calculations of Euler or RANS type. In order to produce surfaces that respond smoothly to changes in the volume function, and avoid an integer optimisation problem, the parameterisation should ideally also use real number volume fractions. Furthermore, as future extensions to multi-level parameterisation are extremely likely, it should also have a refinement capability. The objective for this work is to derive a parameterisation scheme meeting these goals as fully as possible.

2. Volume Fraction Parameterisation

The majority of existing geometry parameterisations used for fluids optimisation either seek to represent the surface directly by building it from splines or deforming an existing geometry. The parameterisation presented here is inspired by the front reconstruction techniques used in the volume of fluid (VOF) method [42] for free surface simulation. In contrast to the VOF method this parameterisation method uses the volume fraction of solid to construct surfaces; the design variables being the volume fractions of solid in the cells making up what is called a parameterisation mesh or grid in this work.

For the VOF method the conservation of fluid volume is of vital importance while adequate results can be produced without smooth surfaces, however when parameterising a geometry for optimisation the volume fraction can be considered merely a convenient design variable and it is not necessary to ensure that any constructed surface strictly achieves the specified volume fraction in the cell. A far more important consideration is the smoothness of the surface, in particular ensuring that it is continuous, a property that the reconstructions used for the VOF method often lack.

The volumetric parameterisation presented in this work is capable of representing both different topologies and geometries, a feature not available in

many parameterisation schemes which can only represent differing geometries with the same topology. The mechanism by which multiple topologies can be represented can be understood by imagining two regions of cells in the parameterisation separated by some distance, a situation illustrated in figure 3. Cells in these regions are set such that they are solid by having a volume fraction of solid equal to one, while the cells separating the two regions have a volume fraction set to zero in order to be completely void. It is evident that two separate surfaces should be constructed. By subsequently changing the volume fraction of solid in the cells, labelled 1 and 2 in figure 3, the two regions may be joined. Figure 4 shows how smooth surfaces may also be constructed when the volume fraction is allowed to vary from zero to one, with intermediate values in this range shown in greyscale, for a two element aerofoil. Figure 5 shows how this approach may also be used to construct the perimeter of a harbour; for example, it might be used in numerical optimisation to find positions and shapes of breakwaters for minimising wave energy within the harbour [52].

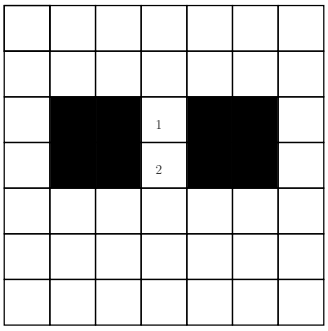


Figure 3: Volumetric parameterisation grid (black represents a region completely filled with solid)

The volumetric parameterisation used here builds on the work of Prilepov *et al.* [53] in the field of computer graphics. Given a grid of cells, each with an associated volume fraction, the method constructs one or more surfaces from the information about the volume fractions in the grid. The steps in producing a

surface from a set of volume fractions can be summarised as follows, (a detailed description follows in section 2.1)

1. Average the cell volume fractions to volume cell vertices.
2. Construct gradients of the volume fraction at the vertices using central differences.
3. Perform an interpolation, using the vertex values of volume fraction and volume fraction gradient, over each volume cell to construct a continuous volume fraction function.
4. In each cell, points where the value of the volume fraction function is higher than a threshold are considered to be within a surface while lower values are outside. For each cell choose the threshold value such that the fraction of the cell area inside the surface is equal to the set volume fraction for that cell.
5. Sample each cell volume fraction function with a Cartesian grid of points to find points within the surfaces by comparing to the cell threshold value.
6. Construct surfaces around interior points using either a marching squares or triangles method.

It should be noted that although this approach is likely to use a relatively high number of design variables compared to the other methods described in section 1.1, scope clearly exists for the use of design space reduction as also described in section 1.1.

Most topology optimisation takes place in terms of structural behaviour [54, 55], so it is worthwhile considering how the VoS method above can be classed in this context. Deaton *et al.* [54] define four types of parameterisation that include topology as (1) density-based, (2) 'hard-kill' methods, (3) boundary variation and (4) cellular division. High-fidelity CFD generally requires boundary fitted meshes, which drives towards using the boundary variation approach, of which

Deaton *et al.* define two types: (a) level-set methods [17] and (b) phase-field methods [56]. Following this terminology, the VoS approach developed here is a phase-field method, but it is apparent that phase-field approaches may also be considered as explicit level-sets. The contribution of this work is therefore a volume based parameterisation capable of sub-cell reconstruction (thereby using fewer design variables) on anisotropic parameter grids, which is suitable for use with boundary fitted fluids meshes, or any methods where the exact boundary is required.

In terms of design space dimensionality the approach developed here is comparable or better than the density based methods employed in structural optimisation, because it is able to construct geometric shapes *within* the parameter cells using an interpolation within each cell. The interpolation uses gradients of the volume of solid function to provide the required level of detail. Achieving improved internal fidelity permits the use of larger parameter cells, correspondingly faster optimisation (due to a smaller number of design variables) and is well suited to fluid-mechanics problems where surface smoothness is particularly important. The cost of using the volume of solid parameterisation is of course higher than the density-based methods used in structural optimisation, because there are moderately intensive steps (primarily mesh edge loops, sampling and then sorting using the volume function) involved in construction of the surface from the volume of solid function and its gradients. The final surface is also built using a marching triangles/squares procedure which carries a small overhead. Certainly for fluids problems, however, the computational cost of the parameterisation is small compared to the physics part of the calculation, which means the benefit of a smaller number of design variables outweighs the penalty of a more costly reconstruction procedure. It would also not usually be possible to use a density-based approach (or in fluids, its analogue, the porosity

method) as the meshes used in fluids analysis are not generally suitable for parameterisation, often featuring highly stretched cells and large numbers of cells.

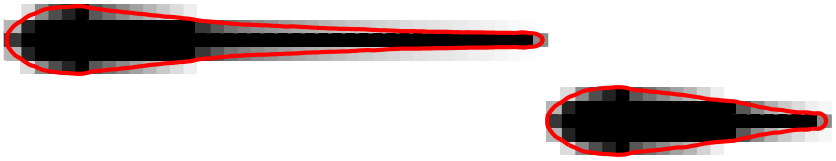


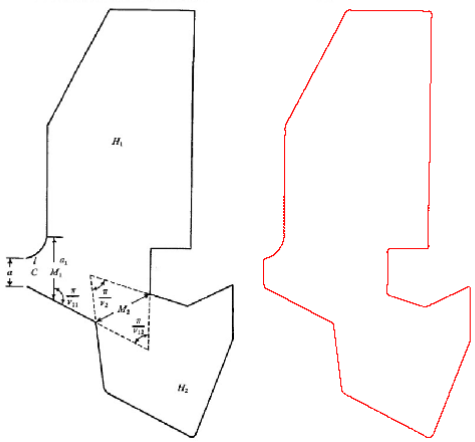
Figure 4: Greyscale plot of underlying volume fraction data and reconstructed surface

2.1. Construction of volume fraction function/gradient

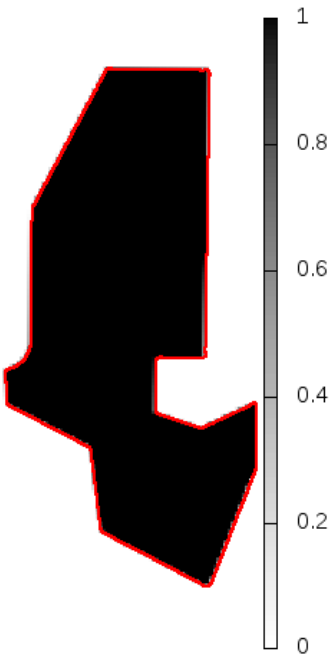
Strictly the volume fraction in a cell defines the fraction of that cell that is taken up by interior volume, however in order to generate smooth surfaces it is essential to have an estimate of the volume fraction at sub-grid scales. This is achieved by interpolating the volume fraction over the cell to construct a continuous volume fraction function, $f(x, y)$, which can be used to calculate whether a point lies within a surface or outside.

First $f(x, y)$ (the volume of solid function) and its gradient are calculated on the cell vertices by averaging the volume fraction for the four surrounding cells to calculate $f(x, y)$, and a volume integral is used to approximate the gradient (a detailed description of the process is presented later, in section 2.2). Figure 6 shows an example problem with cell volume fractions generated from a real interface and corner averages and gradients calculated for the central cell.

This information allows cubic Hermite splines [58] to be fitted to each cell edge to interpolate $f(x, y)$ along each cell edge following equation (1), where the Hermite polynomials H_i^3 are given by equations (2) to (5). p describes the value of the function to be interpolated, $\mathbf{m}$ is the gradient, $\mathbf{l}$ is the vector along the edge and t is defined over the interval $[0, 1]$ such that $p_1 = p(1)$.



(a) Target geometry (left) and volume of solid constructed contour (right)



(b) Volume of solid contour overlaid on VoS greyscale

Figure 5: Reconstruction of Long Beach Harbour [57], California

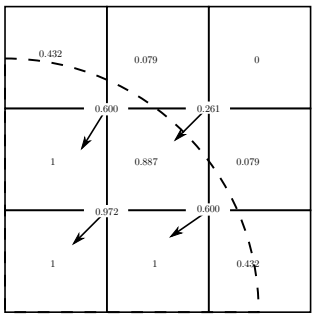


Figure 6: Real interface, cell volume fractions (central values), corner averages (values on the corners) and gradient approximations (arrows, not shown to scale)

$$p(t) = p_0 H_0^3(t) + (\mathbf{m}_0 \cdot \mathbf{l}) H_1^3(t) + (\mathbf{m}_1 \cdot \mathbf{l}) H_2^3(t) + p_1 H_3^3(t) \tag{1}$$

$$H_0^3(t) = (1-t)^2(2t+1) \tag{2}$$

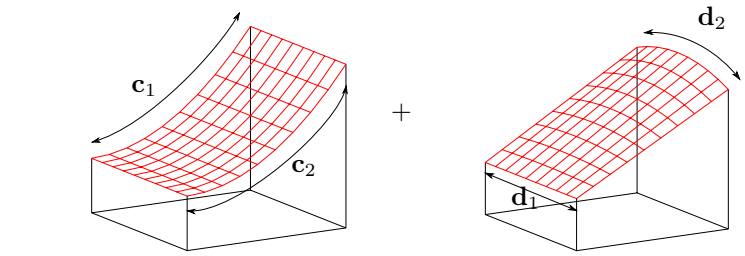
$$H_1^3(t) = (1-t)^2 t \tag{3}$$

$$H_2^3(t) = t^2(t-1) \tag{4}$$

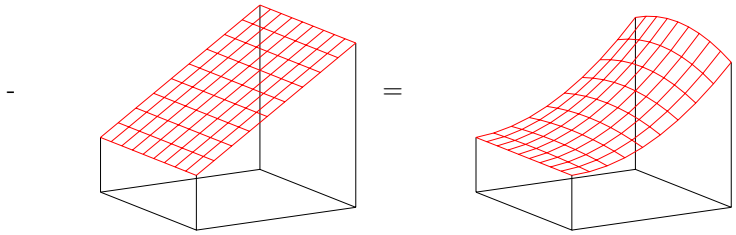
$$H_3^3(t) = t^2(3-2t) \tag{5}$$

Finally $f(x, y)$ is calculated over the whole cell using a bilinearly blended Coons patch [59] using the edge interpolations. Importantly this process produces an independent interpolation for each cell and does not require the solution of a large system of equations. An example of this method being used for a cell in the reconstruction scheme is shown in figure 7, note particularly how each component of the Coons patch behaves at the bounding curves. For a surface bounded by four curves, figure 7d, a Coons patch is created by generating two ruled surfaces, $\mathbf{r}_c$ between bounding curves $\mathbf{c}_1$ and $\mathbf{c}_2$, figure 7a, and $\mathbf{r}_d$ between bounding curves $\mathbf{d}_1$ and $\mathbf{d}_2$, figure 7b. A bilinear interpolation between the four corners, $\mathbf{r}_{cd}$, is also generated, figure 7c, and the Coons patch is then given by

equation (6) which takes the best features from all three interpolations. Using the Coons patch combined with the cubic interpolation along each cell edge allows multiple crossings of the function over each edge and maxima and minima to occur within the cell rather than at a corner, allowing a larger number of geometries to be represented when compared to lower order interpolations.



(a) The ruled surface $\mathbf{r}_c$ (b) The ruled surface $\mathbf{r}_d$



(c) The bilinearly interpolated surface $\mathbf{r}_{cd}$ (d) The Coons patch $\mathbf{r}$

Figure 7: Ruled and bilinearly interpolated surfaces generated by the reconstruction scheme

$$\mathbf{r} = \mathbf{r}_c + \mathbf{r}_d - \mathbf{r}_{cd} \tag{6}$$

2.2. Gradient and Corner Averaging Details

If the parameterisation grid is made up of uniform cells then the cell corner values of the volume fraction and volume fraction gradient can be calculated by simple arithmetical averages and finite differences. However, here the parameterisation cells are allowed to be anisotropic.

Clearly the volume fraction in a cell, defined on the cell centroid, should be averaged to the cell corners in a way that respects the varying distances from the surrounding cell centroids to the corner point. The method chosen for this is an interpolation scheme called inverse distance weighting (IDW) [60]. This scheme interpolates a function by means of a weighted average where the weightings are the inverse distance from the evaluation point to the data point. The form of the interpolation is shown in equation (7) where N is the number of adjacent cells, either three or four, and $\mathbf{x}_i$ is the location of an adjacent cell centroid. In this case $\|\cdot\|$ is the Euclidean norm and p controls how rapidly the change between data points occurs, here it is set $p = 2$.

$$f(\mathbf{x}) = \begin{cases} \dfrac{\sum_{i=1}^{N} \frac{f(\mathbf{x}_i)}{\|\mathbf{x}-\mathbf{x}_i\|^p}}{\sum_{i}^{N} \frac{1}{\|\mathbf{x}-\mathbf{x}_i\|^p}} & \|\mathbf{x}-\mathbf{x}_i\| \neq 0 \\ f(\mathbf{x}_i) & \|\mathbf{x}-\mathbf{x}_i\| = 0 \end{cases} \tag{7}$$

If the cells form a uniform grid and there is no possibility for individual cells to be refined, simple finite differences can be used to calculate the gradients at the cell nodes. However, with the additional complication of a variable number of cells neighbouring a corner and when the variable distance to cell centroids is introduced this approach is not suitable. Instead, the gradient is calculated at the cell centre using a finite volume scheme and these gradients are then averaged to the corners by the same process used for the volume fraction itself.

The finite volume scheme calculation of the gradient at the centre of cell i can be found by first considering the gradient of the volume fraction function $\nabla f(\mathbf{x})$ as

$$\nabla f_i \approx \frac{1}{A} \int \nabla f(\mathbf{x}) d\Omega \tag{8}$$

where $d\Omega$ is a differential volume. This can be transformed to the surface integral

$$\nabla f_i \approx \frac{1}{A} \int f(\mathbf{x}) \mathbf{dS} \tag{9}$$

where **dS** is a differential surface on the boundary of $d\Omega$. This integral can be approximated with a summation

$$\nabla f_i \approx \frac{1}{A} \sum_{j=1}^{N} l_j 0.5(f_i + f_{\text{neighbour}})\mathbf{n} \tag{10}$$

where l_j is the length of the j'th edge, N is the total number of edges and **n** is the face unit normal vector. Note that a side of a cell may be made up of more than one edge if more than one cell borders the cell on that side. These gradients are then nodally averaged and used within the Hermite interpolation.

2.3. Selecting Iso-value

The choice of whether a given value of $f(x, y)$ represents a point inside a surface or not is made by defining a per cell clipping value σ, or equivalently a value of the contour of $f(x, y)$, that represents the surface. This value should be chosen such that the fraction of the cell volume within the surface contour corresponds as closely as possible to the given volume fraction in the cell. Figure 8 shows this diagrammatically.

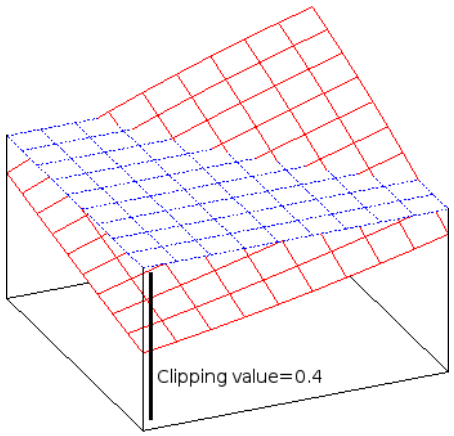


Figure 8: Curved volume fraction function and clipping plane. Points where the volume fraction function is greater than the clipping plane are inside a surface. This figure corresponds to a cell clipping value of 0.4

The value of $f(x, y)$ is densely sampled in each cell on a Cartesian sample grid and the value of $f(x, y)$ at these points is sorted. If N_s is the total number of points sampled the largest n samples are selected where n/N_s is equal to the volume fraction in the cell, σ is then set to be the lowest of the n selected sample values. This process is shown in figure 9.

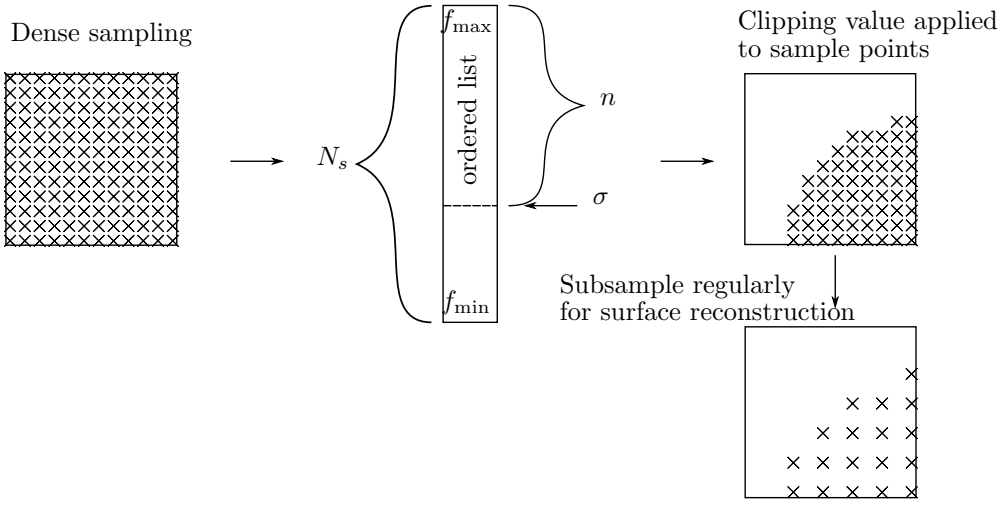


Figure 9: The process of selecting a clipping value

2.4. Smoothing Iso-value

As the value of σ is defined on a per cell basis the join between the set of points within a surface in one cell and the set in an adjacent cell may be non-smooth due to the different values of the surface contour across a boundary. This can be solved by sacrificing some volume accuracy by smoothing the values of σ across cell boundaries such that $\sigma = \sigma(x, y)$ and is no longer cell-wise constant. Any method should have a number of key properties:

- Only nearby cells contribute to the value of σ
- The degree of smoothing must be controllable
- On a cell boundary the contribution from adjacent cells should be equal

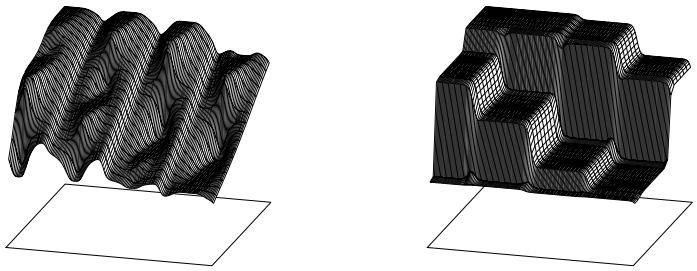
- The function must provide smooth changes
- Differently sized and anisotropic cells must be correctly handled
- The smoothing method must be able to cope with arbitrary data point locations

The method of IDW described in section 2.2 provides the basis for the method employed. Simply, IDW can be applied with the value of the distance exponent p, defined in equation (7), set to some large value, typically around 10-20, in order to effectively exclude far away cells from affecting the smoothed value. However this simple method encounters difficulty with anisotropic cells or indeed cells of differing sizes as the contribution from cells adjacent to an edge would not be equal at the edge.

Even in the case of equally sized cells problems can appear with a simple IDW approach, and more so in the case of anisotropic cells. The smoothed σ surface can be highly oscillatory on the borders between cells with different cell centre values of σ, this phenomena can be seen in figure 10a. This occurs due to the anisotropy of the cells and the use of high powers of distance in the IDW weights. This oscillatory behaviour in the smoothed value of σ causes oscillatory surfaces to be constructed when the surface crosses a cell boundary.

Two modifications are applied to make IDW suitable for this smoothing application. Distances are not measured from the cell centroid to the point at which σ is evaluated but rather from a rectangular box centred on the cell centroid to the evaluation point. This box is smaller than the cell it lies within and is sized such that the perpendicular distance from cell edge to the box edge is constant around the whole cell. Measuring distance as between the box and the evaluation point ensures both that each cell has a region of the same size where smoothing takes place and that the central part of the cell is set to the cell centroid value of σ, ensuring smoothing only takes place at cell edges. Figure 11

illustrates the concept of the inner box.



(a) Simple IDW method (b) New IDW method

Figure 10: Comparison of smoothed σ function generated with different smoothing methods

Additionally, the Euclidean distance is not used but rather the Chebyshev distance, defined in two dimensions by equation (11) where **i** and **j** are orthogonal vectors corresponding to the axis directions. The preference for the Chebyshev distance over the Euclidean distance is due to the behaviour of circles when defined using each distance (when using the Chebyshev distance a circle appears as a square). This means that in any cell the distance from the inner box to the cell boundary is constant, implying equal contribution from adjacent cells on cell boundaries.

$$\|\mathbf{x} - \mathbf{x}_i\| = \max(|(\mathbf{x} - \mathbf{x}_i) \cdot \mathbf{i}|, |(\mathbf{x} - \mathbf{x}_i) \cdot \mathbf{j}|) \tag{11}$$

Figure 10b shows a smoothed σ function generated with this method. Note that smoothing takes place over the same distance irrespective of cell size and that smooth surfaces are generated. Figure 12a shows the oscillatory surface generated with the simple IDW method for smoothing σ while figure 12b demonstrates the improvement obtained with the new method for the same input data.

Alternative approaches can be trialled for smoothing the clipping function, for example, it is also possible to use a RBF interpolation that fits through all of the parameter cell clipping values. A comparison between the IDW approach

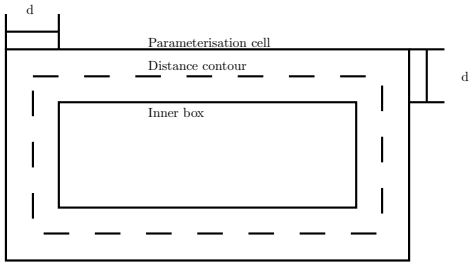
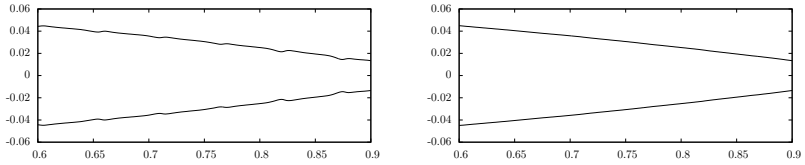


Figure 11: Schematic of distance measurement showing the central box, a dashed distance contour and the outer volume cell

described above and an RBF approach is shown in figure 13, where it is clear that the RBF approach is not as suitable, as the contour produced is less smooth. This is attributable to the RBF interpolation capturing local cell-wise variations in the clipping value more faithfully. At first sight this might appear desirable, but the intention of the surface construction is not precise recovery of either clipping values or volume of solid values; rather, it is important that a smooth surface that responds predictably to changes in the volume of solid values is produced. In this sense the IDW interpolation of the clipping value is superior.



(a) Simple IDW method (b) Proposed IDW method

Figure 12: Comparison of surfaces generated with different smoothing methods

2.5. Constructing Surfaces

To construct the surfaces a sampling of points in each cell is taken to determine which of the sample points lie within a surface by comparing the value

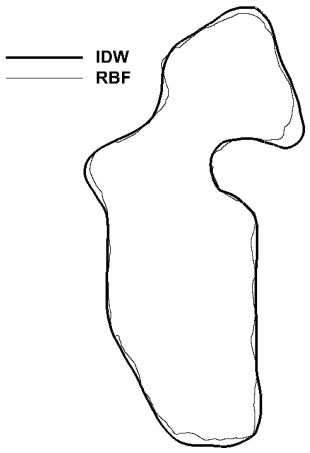


Figure 13: Comparison of surfaces generated using the proposed IDW and RBF interpolations for clipping value

of $f(x, y)$ with the smoothed value of $\sigma(x, y)$ at each point; this produces a structured point cloud of internal points. This process is shown in figure 9 for an unsmoothed value of σ. Surfaces can then be extracted from the set by the method of marching squares [61] or the related method of marching triangles. Internal points and a corresponding reconstructed surface are shown in figure 14 for a low sample density.

The number of sample points can be varied depending on the desired smoothness of the resulting surfaces, which will be problem dependent. For instance, for minimising drag on an object in a supersonic flow where the objective function is to be evaluated by an Euler solver it is likely that the optimal geometry will feature sharp corners in order to produce attached shocks. Setting the number of samples per cell to be one will produce surfaces with such sharp features; figure 15a shows such a geometry, while a higher number will yield smoother surfaces as seen in figure 15b.

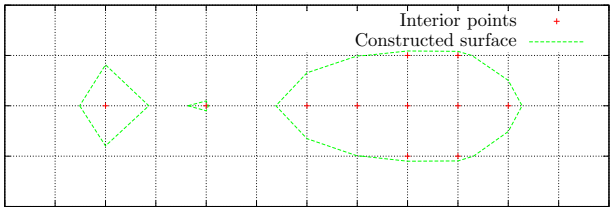


Figure 14: Interior points and surface generated using marching squares algorithm

(a) Example geometry produced with one sample per cell

(b) Example geometry produced with 400 samples per cell

Figure 15: Surfaces produced with different numbers of sample points but the same volume fraction distribution

Marching squares [61] is an excellent method where the sampling of the volume fraction is uniform across the parameterisation mesh but it does not allow the flexibility of having differing sample densities for different regions, necessary for localised control of output surface smoothness. Instead a Delaunay triangulation is constructed using the sample points as cell nodes. This triangulated mesh is then used to construct the surface by marching triangles, a method which applies the same principles as marching squares but is suitable for less ordered data and removes the ambiguities which exist in surface construction

in the marching squares method.

The tendency of the generated surface to split into separate surfaces or form a cohesive whole depends on the gradient of the volume fraction function. When the gradient is high, especially if the gradient is in the same directions at either end of an edge, the Hermite splines that interpolate the function will tend to have maxima and minima within the bounds of the edge causing the generated surface to break up. Figure 16 shows two splines generated with and without the gradient control described here. If a clipping value was applied to these functions around $y = 1$, the uncontrolled spline would have two regions where sample points would be considered within the surface whereas the controlled spline would have only one. This demonstrates how the scaling value can affect how cohesive the generated surfaces are. By scaling the gradient in each cell by a constant factor, the degree of cohesion can be controlled and figure 17 demonstrates the differences in the generated surface that can be obtained.

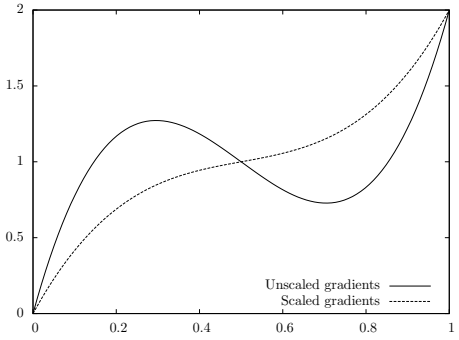
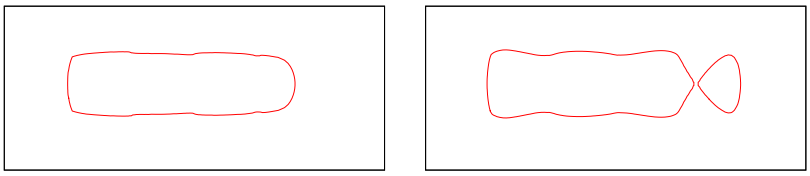


Figure 16: Two Hermite splines, given by equation (1), illustrating influence of gradient control

2.6. Sharp Corners

By taking fewer samples the smoothness of the reconstructed surface is reduced as the straight line segments generated by the marching triangles method become longer. This can be useful when considering the design of bodies which are likely to need to be angular, such as for supersonic flow. The marching



(a) Small scaling factor producing single lumped surface

(b) Large scaling factor producing two broken surfaces

Figure 17: Varying gradient scaling and resultant surfaces

triangles surface construction method allows for local control of the sampling density on a per cell basis which enables local control of the generated surface smoothness. This is particularly important when designing geometry such as aerofoils, as a sharp trailing edge is often a requirement.

Having dramatic differences between sample density of adjacent cells is only possible with use of marching triangles on an appropriately constructed triangular mesh, in this case by Delaunay triangulation. When a sharp trailing edge is required, a cell is flagged to contain the trailing edge and this cell is sampled according to the pattern shown in figure 18, though sample locations are actually closer to the edges of the parameterisation cell in practice. In this case a constrained Delaunay triangulation is used for the marching triangles mesh, such that the lines joining the trailing edge cell sample points are forced to be in the triangulation. Figure 19 shows how forcing this segment to appear in the triangulation ensures a sharp trailing edge with the point located on the line joining the sample locations.

3. Results

3.1. Comparison with other Parameterisations

A standard test [62, 40] of a parameterisation scheme focusses on the capability to successfully recover an existing aerofoil geometry using as few design

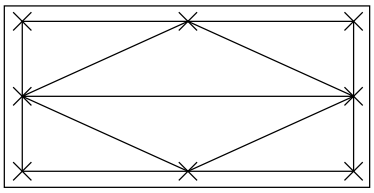


Figure 18: Sample locations (crosses) in a parameterisation cell with enforced triangulation edges connecting the sample points

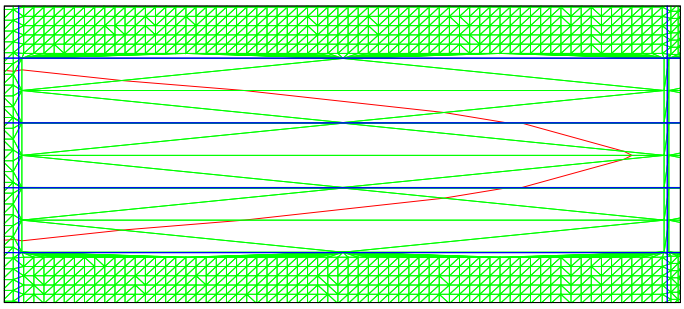


Figure 19: Use of a constrained Delaunay triangulation to ensure a sharp trailing edge. Parameterisation cells in blue, triangulated sample mesh in green and constructed surface in red

variables as possible. Figure 20 shows the method used here for shape recovery, which is based on iteratively matching the measured volume fractions in the parameterisation grid between the target surface and the parameterised surface to minimise the RMS error in equation (13).

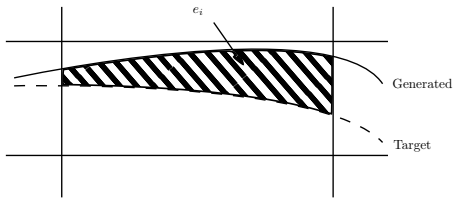


Figure 20: Volume fraction error

Many of the methods used to parameterise aerodynamic shapes are either formulated specifically for that class of shape or instead are deformative methods that use an initial aerofoil shape which is then deformed to form new geometries. The volumetric parameterisation method presented here is more general and capable of representing many geometries. The more specialised methods have, in a sense, more information about what an aerofoil looks like embodied into the scheme; in principle this should allow these parameterisations to represent the target aerofoil with fewer design variables to the same error value.

Castonguay and Nadarajah [63] compare different parameterisation methods for the optimisation problems of drag minimisation and inverse design on two dimensional aerofoils, while Mousavi *et al.* [64] extend this work to the three dimensional problem of drag minimisation and inverse design for wings. Here the parameterisation is examined in the context of shape recovery, alongside the work of Masters *et al.* [41], who compared a large number of geometry parameterisation methods and evaluated the number of design variables necessary to reproduce, within a tolerance specified by Kulfan [7], fractions of a large library of aerofoils. In order to compare to the results of Masters *et al.* [41] the error here is calculated after a resplining procedure using an identical method, after

which the RMS error is calculated following equation (13) where N is the number of generated surface points, $y_t(x_i)$ is the y coordinate of the point on the target surface at x_i and $y(x_i)$ is the point on the generated surface.

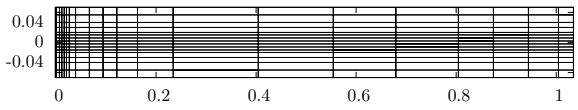
$$x_i = \frac{1 - cos(\pi(\frac{i-1}{150} - 1))}{2}, \quad 1 \leq i \leq 301 \tag{12}$$

$$\min \sum_{i=1}^{N} \sqrt{\frac{|y_t(x_i) - y(x_i)|^2}{N}} \tag{13}$$

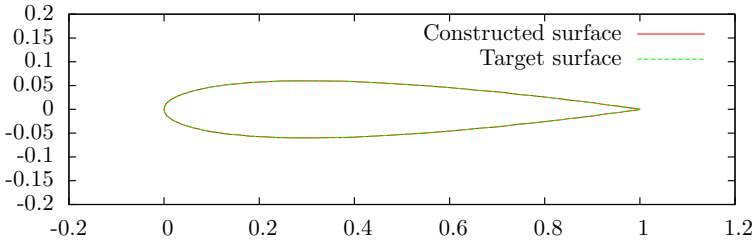
A more detailed analysis can be performed by considering the front 20% of chord and the rear 80% of chord separately and identifying the furthest distance a constructed point in each region is from the target surface. This allows these errors to be compared to a wind tunnel model tolerance which is given as 4×10^{-4} for the front region and 8×10^{-4} for the rear region of the chord [41] (if the aerofoil is recovered within these tolerances then it can be considered a successful recovery). This tolerance comes from the work of Kulfan [7] where it was suggested as a value indicative of wind tunnel model manufacturing accuracy.

NACA 0012. For the NACA 0012 reconstruction a grid of 20 cells horizontally by 15 cells vertically, shown in figure 21a, was used. More resolution was concentrated around the leading edge especially and slightly more around the trailing edge. The mid-chord region has lower resolution by comparison as the curvature here is low. As stated earlier, for optimisation cases only the parameterisation cells on the surface of the aerofoil are relevant to the optimisation process and should be counted as design variables; in this case there were 64 of these cells.

Figure 22a shows the qualitative matching between the geometry constructed by the parameterisation and the target surface. The maximum error in the front 20% of chord is 3.99×10^{-4} while in the rear 80% it is 5.96×10^{-4}.



(a) Volume fraction mesh for NACA 0012 shape recovery



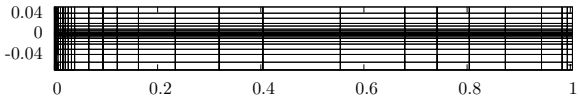
(b) Comparison between generated aerofoil and target surface for the NACA 0012

Figure 21: NACA 0012 section parameterisation mesh and surface comparison

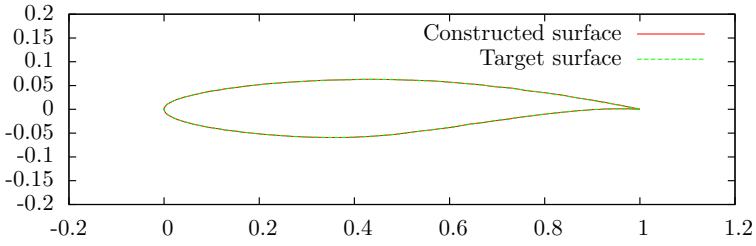
RAE 2822. More parameterisation cells are required to represent the RAE 2822 to the same tolerance as the NACA 0012 owing principally to the thinner leading and trailing edge geometry. The parameterisation mesh is shown in figure 22a and is made up of 26 cells horizontally by 21 cells vertically. For this geometry and parameterisation grid the number of surface cells, or design variables for optimisation, is 95. When comparing the NACA 0012 result above with this result it can be seen that the number of design variables in the RAE 2822 case is larger than in the NACA 0012 case by a factor of 1.48 but the total number of cells in the parameterisation mesh is larger by a factor of 1.82. This indicates that with the increased resolution the total number of cells in the mesh will increase faster than the number of cells that affect the surface geometry. This is due to the total number of cells increasing with the square of the number of cells on a side, while active cells are approximately proportional to the total arc length.

The parameterisation produces a geometry that meets the error tolerance with a maximum error of 3.99×10^{-4} in the front 20% and 7.42×10^{-4} in the rear 80%. The comparison between the generated surface and the target is shown in figure 22b.

The volume of solid (VoS) parameterisation requires 95 design variables to meet the Kulfan tolerance while most other methods examined by Masters *et al.* [41] require between 10 and 20 design variables. However, this is to be expected when the other methods are examined. These methods either deform an initial aerofoil geometry or are specially formulated to produce aerofoil like geometries whereas the VoS parameterisation presented here is designed to be applicable to problems other than aerofoil optimisation and is also capable of topological change.



(a) Volume fraction mesh for RAE 2822 shape recovery



(b) Comparison between generated aerofoil and target surface for the RAE 2822

Figure 22: RAE 2822 parameterisation mesh and surface comparison

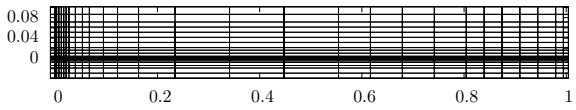
NACA 4412. The NACA 4412 is a more highly cambered aerofoil than any of the others examined here, this requires the parameterisation mesh, shown in figure 23a, to be substantially more asymmetric than even the RAE 2822 case. More parameterisation cells are also required, 28 cells horizontally by 21 vertically, with most resolution distributed about the leading and trailing edges. The number of surface cells acting as design variables is 96 in this case and the maximum errors are 3.97×10^{-4} in the front 20% of chord and 7.57×10^{-4} in the rear 80% of chord, but just below the defined tolerance.

ONERA D Section. Like the NACA 0012 the ONERA D section is a symmetrical aerofoil but with slightly lower thickness. This results in higher curvature at the leading edge, and so more resolution is needed for reconstruction compared to NACA 0012. Figure 24a shows the parameterisation mesh used for this reconstruction which was formed from 23 horizontal cells by 17 vertical cells, with 76 of those cells being on the surface and acting as design variables.

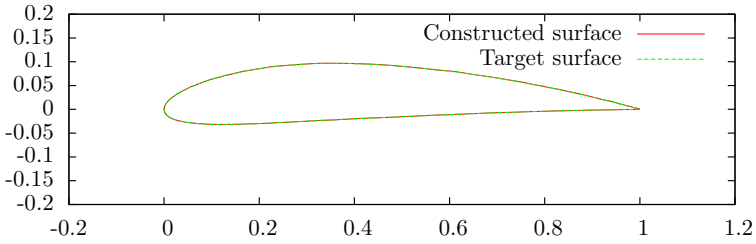
For this geometry reconstruction the generated surface, shown in figure 24b, has a maximum error of 3.66×10^{-4} in the front 20% of chord and 7.74×10^{-4} in the rear 80% of chord. Again, these values are below the specified target tolerance.

In comparison to the NACA 4412 and the RAE 2822, the VoS scheme is closer in performance to the other parameterisation methods when reconstructing the ONERA D section as other methods require slightly higher numbers of design variables on average while the VoS parameterisation needs fewer for this geometry. This is due to the D section being symmetrical, meaning fewer parameterisation cells can be used around the trailing edge as there is no requirement to reconstruct a concave or thin trailing edge.

Results Summary. The results in the above sections are summarised in table 1 which illustrates that some geometries are harder for the VoS parameterisation

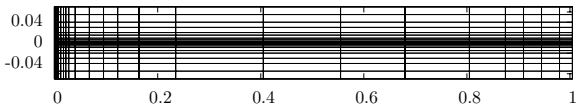


(a) Volume fraction mesh for NACA 4412 shape recovery

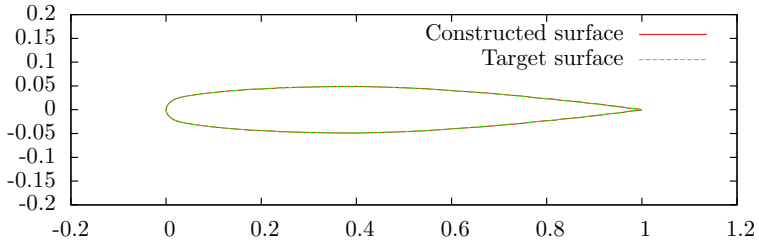


(b) Comparison between generated aerofoil and target surface for the NACA 4412

Figure 23: NACA 4412 parameterisation mesh and surface comparison



(a) Volume fraction mesh for ONERA D section shape recovery



(b) Comparison between generated aerofoil and target surface for the ONERA D section

Figure 24: ONERA D section parameterisation mesh and surface comparison

Aerofoil	Front error	Rear error	Number of design variables
NACA 0012	$3.99 \times^{10} -4$	$5.96 \times^{10} -4$	64
RAE 2822	$3.99 \times^{10} -4$	$7.42 \times^{10} -4$	95
NACA 4412	$3.97 \times^{10} -4$	$7.57 \times^{10} -4$	96
ONERA D section	$3.66 \times^{10} -4$	$7.74 \times^{10} -4$	76

Table 1: Table of errors and design variable numbers required to reconstruct a selection of aerofoils to a tolerance specified by Kulfan [7]

to recover than others. The number of necessary design variables varies by a factor of 1.5, and symmetric aerofoils are more easily recovered, though this is likely not a factor of the symmetry of the itself but rather due to the concave regions on the lower surfaces of the asymmetric aerofoils.

3.2. Supersonic Minimum Drag Bodies

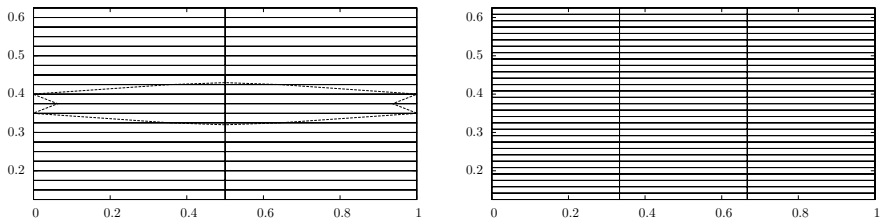
The Busemann biplane [65] is a theoretically zero drag configuration, in an inviscid flow, due to wave cancellation between the bodies making up the biplane. This effect relies on two bodies to create the wave cancellation and as such this design would not be available to a parameterisation scheme which is not sufficiently flexible to perform topology optimisation as well as geometry optimisation without the initial assumption of two bodies.

Optimisation of Busemann biplanes has been performed before using other parameterisation schemes [11, 66] but these optimisations have been aimed at improving the performance of an already existing biplane configuration, typically to improve performance when the biplane is used as a lifting body at non-zero angle of attack. In this work, the possibility of using a parameterisation scheme capable of topological change to harness the benefits of shock cancellation between bodies to reduce drag is explored.

Initially the parameterisation grid is set up as in figure 25a, comprising of two cells horizontally and twenty vertically though the volume fraction values are mirrored around the horizontal line splitting the mesh in half such that there are twenty design variables and any design is symmetric around the horizontal,

though not necessarily around the vertical. The marching squares surface construction method is used along with a low sample density in the parameterisation grid in order to force the design to be formed of sharp, faceted bodies.

The objective function is to minimise drag on the body in a Mach 2 flow, subject to a constraint on volume such that the total volume of all bodies produced must be greater than 8% of the volume of the parameterisation mesh; this prevents the optimiser from producing a minimum drag design by removing all objects in the flow. The optimisation algorithm used is the gravitational search algorithm (GSA) [67] as a gradient based method would find it difficult to produce good results due to the possibility of better designs being hidden behind increases in the objective function associated with changes in geometry and topology. The GSA method was previously tested against the CEC 2006 optimisation test functions [68], as well as three engineering test cases (welded beam [69], pressure vessel [70] and spring design [71]) and found to provide convergence and feasibility results of similar or better quality than comparable agent-based methods.



(a) Low resolution parameterisation grid (b) High resolution parameterisation grid and initial geometry

Figure 25: Parameterisation grids

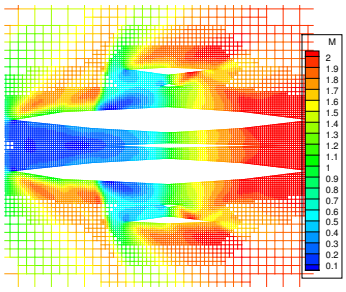
The optimisation produced a 96.7% reduction in the objective function, almost entirely eliminating the drag on the body. The final geometry and the

flow field around it can be seen in figure 26f where the optimal geometry that has been found is formed of five separate bodies. Figure 26 shows some designs from the optimisation history which demonstrates that a number of topologies were considered in this optimisation. The mechanism of drag reduction seen here is exactly that utilised in the theoretical Busemann biplane, namely shock cancellation, which can be seen by the minimal trailing waves produced. The observed flow field is more complex than that of the theoretical biplane result as each shock wave undergoes multiple reflections which are not necessarily focused on the maximum thickness points of adjacent bodies.

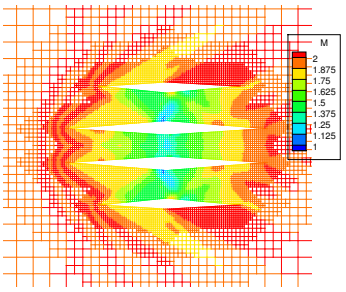
Given the incomplete cancellation in the solution produced using the parameterisation grid shown in figure 25a, better results might be obtained by using a higher resolution parameterisation grid such as that in figure 25b where the resolution has increased by a factor of 1.5. All other parameters for the optimisation remained the same as with the lower resolution parameterisation.

The increased parameterisation resolution has the effect of enabling the optimiser to reach a reduction in the objective function of 99.5%, in fact this is a reduction in the drag of nearly an order of magnitude over the result obtained with the lower resolution parameterisation. Once again, the selected design iterations shown in figure 27 show that a number of different topologies have been considered in the optimisation process. The final geometry, seen in figure 27f features more, thinner bodies leading to better shock cancellation and a reduction in the shocks on the outside edges of the top and bottom aerofoils which cannot be cancelled.

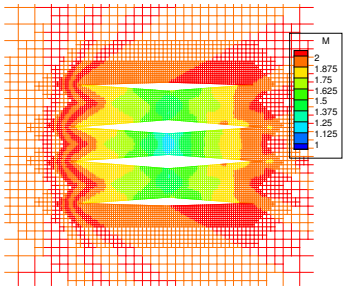
The histories of the optimisations for both the coarser and finer parameter meshes are shown in figure 28. The finer parameter grid ultimately achieves a lower value of C_D, but the convergence to this value takes longer in terms of iterations of the optimiser. It would be expected that the finer parameter



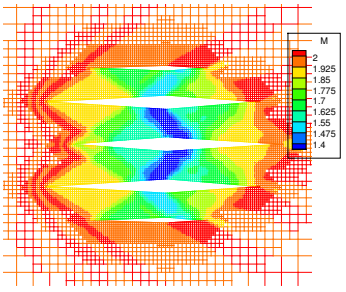
(a) Initial configuration



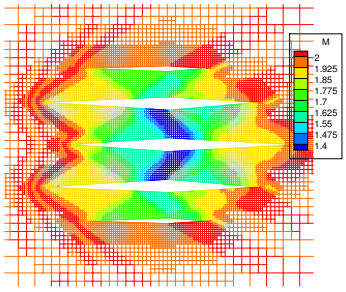
(b) Iteration 138



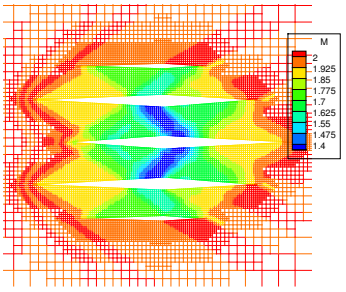
(c) Iteration 276



(d) Iteration 414



(e) Iteration 552



(f) Final configuration

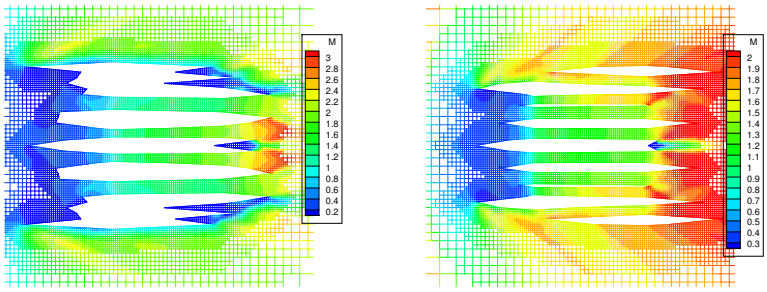
Figure 26: Flow field around design iterations. Low resolution parameterisation

grid achieves lower drag, as (i) it has an increased number of design variables to adjust, and (ii) with a finer grid it is possible to construct a wider range of topologies, as evidenced by the increased number of elements to the aerofoil system. A smaller number of design variables, as for the coarser parameter grid, allows more rapid convergence of the optimiser but the quality of the final result is limited as fewer shapes are achievable.

The classical zero drag Busemann solution is formed of two bodies, however the two solutions presented above feature a greater number than that while still achieving low drag configurations. One possible reason for this is that there are numerous topologies that achieve cancellation and once one of these designs is reached, the optimiser has no reason to alter the topology further. While there may be many topologies that produce shock-expansion cancellation (for example, each parameter grid above produced a different topology), the fact that the optimiser located these solutions at all is encouraging and indicates that the GSA method has functioned successfully in terms of initial seeding of points and enforcement of constraints. The designs feature near-exact wave cancellation (to within a margin of numerical error), so a global minimum has almost certainly been located, albeit one that is evidently non-unique due to the multitude of ways in which the waves can be made to cancel and produce zero drag.

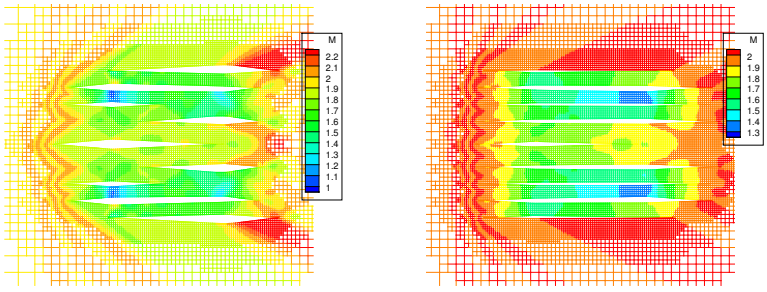
3.3. Coastal Defence Optimisation

A common test case problem in the smoothed particle hydrodynamics (SPH) community is the dam break, where pressures and front position are compared with experimental data or the results of other numerical methods [72, 73]. A problem of interest to any community located close to the shore is also how to prevent wave overtopping of seawalls where infrastructure is located just behind the seawall; for example, preventing nearby buildings being flooded or



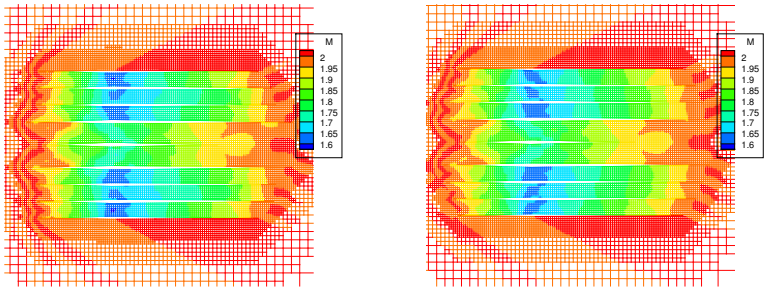
(a) Initial configuration

(b) Iteration 125



(c) Iteration 250

(d) Iteration 375



(e) Iteration 500

(f) Final configuration

Figure 27: Flow field around design iterations. High resolution parameterisation

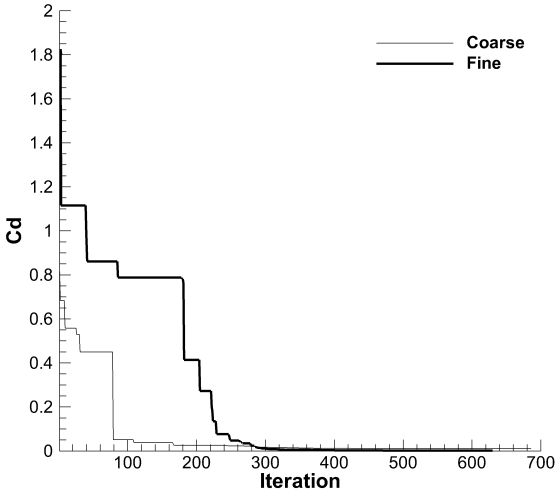


Figure 28: Comparison of optimiser histories for the Busemann problem with coarse and fine parameter grids

damaged by the force of impacting water. The key objective is to stop as much water as possible overtopping the coastal defence; here this concept is expressed mathematically as minimising the total amount of fluid mass crossing a vertical datum line from the seaward side to the coastal side over a set period of time, as given in equation (14) where $\delta()$ is the Dirac delta function. The problem is shown graphically in figure 29 where the fluid initial configuration and the area available for design can be seen. The design area is split into a grid of square volume cells, two cells wide and four cells high, which gives eight design variables.

$$\min \int_{t=0}^{t=t_{\text{end}}} \left(\sum_{i=1}^{N} m_i \frac{\max(V_x, 0)}{V_x} \delta(x - x_{\text{datum}}) \right) dt \tag{14}$$

The optimisation of the objective function given by equation (14) is performed by a conjugate gradient method without any constraints applied to the optimisation. However, constraints on the geometry are applied within the pa-

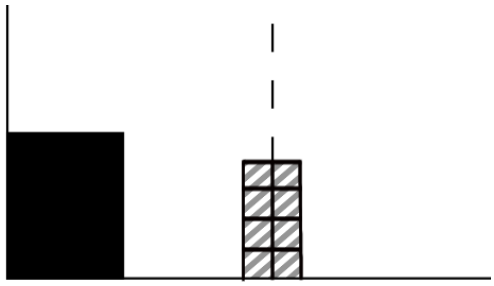


Figure 29: Seawall overtopping optimisation problem. Tank boundaries shown as solid lines, initial fluid position as solid rectangle, potential design area as hatched rectangle and the datum location as a dashed line.

rameterisation as this limits the maximum width and height that the coastal defence can be. Additionally, the parameterisation grid has cells under the horizontal, which do not act as design variables, set to a value of 1 to ensure the generated geometry is always anchored to the x-axis. Implementing the conjugate gradient method requires a line search along the direction calculated for exploration, but this requires some modification to work with the VoS parameterisation. The volume of solid values cannot fall below zero or exceed one, so it is necessary to clamp any design variable that would otherwise exceed those limits at either zero or one as appropriate. It would not be prudent to move to the next parameter cell (although this is seemingly an attractive option), as this would move too far beyond the base configuration for which the gradients were calculated at that step. This is a the 'projection' approach of Rosen [74], and in the context of the work here, a primary consequence is that monotonic convergence is not guaranteed, because predicted search directions must be somewhat truncated to be allowable within the parameterisation.

The results of this optimisation suggest that geometry optimisation using SPH is both possible and yields excellent results, especially when the ease with which the SPH solver can be integrated with a parameterisation that produces changing topology is considered. The final geometry produced is shown, in

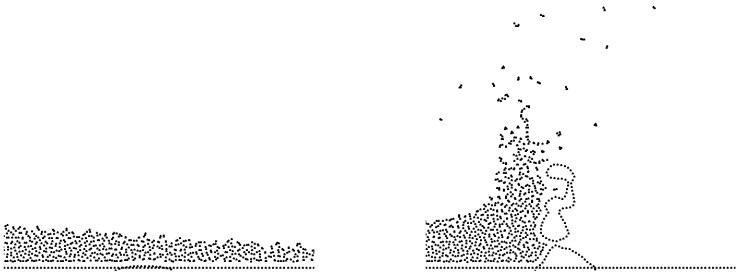
figure 30f, undergoing an impact with the water from the dam break. The optimiser has produced a curved frontal surface to the seawall which redirects the wave away from the datum line and back on itself and this is a feature common to most of the designs produced in the optimisation process, as shown in figure 30. The rear of the seawall is curved despite not interacting with the oncoming fluid. This is an artefact of the volumetric parameterisation; in this case as the volume cells are only two wide the values of the design variables in the front layer of cells affects the surface constructed in the rear layer and vice versa. Figure 31 shows the reduction in the objective function with iteration number; in a relatively low number of iterations the objective function is reduced to zero. Some non-monotonic convergence is seen, attributable to the projection process for the design variables discussed above.

3.4. Baffle optimisation

The selection of internal baffle geometry has the potential to alter the damping characteristics of an oscillating tank. Anderson *et al.* [75] show that the damping performance of an oscillating tank can be improved by the introduction of baffles, though correct placement is essential for maximum effect. In that context, an optimisation method for the design of tank baffles to improve damping should offer the ability to design the number, placement and geometry of the baffles in a given tank.

The volumetric parameterisation method presented in this work is ideal for this application because by allowing for variable topology the number and placement of the baffles can be optimised while also allowing the optimisation of the individual baffle geometry, all within a continuous optimisation procedure.

The geometry chosen to contain the baffles and the fluid is a square tank with total area of 0.25m^2 pivoted about a point 0.115m from the center of the top of the tank in the direction perpendicular to the tank top, as used by Hall



(a) Initial configuration

(b) Iteration 5

(c) Iteration 10

(d) Iteration 15

(e) Iteration 20

(f) Final iteration

Figure 30: Design iterations of the seawall

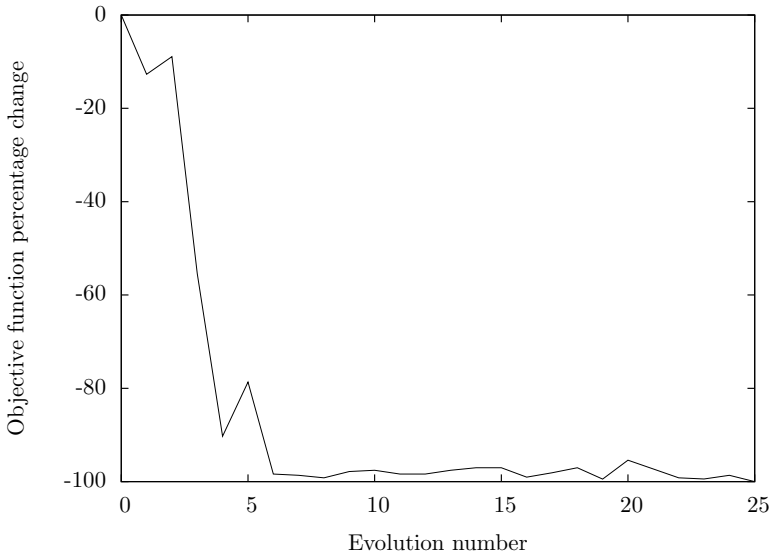


Figure 31: Objective function percentage change against iteration number for the seawall problem

et al. [76]. The initial angle of the tank to the vertical is 20° and the tank is a quarter full. The objective function to be minimised by the optimiser is the total variation of angular position given by equation (15). Here t_2 is five seconds after the tank is released and ω is the angular velocity of the tank. No damping is applied to the tank pivot so any damping experienced is purely due to the fluid inside the tank.

$$F = \int_{t_1}^{t_2} |\omega| dt \tag{15}$$

The SPH solver is coupled to a structural solver which calculates the position of the tank under the forces imposed by the fluid, and the conjugate gradient optimiser is used to drive the design variables. The SPH solver and the structural solver are strongly coupled by sub-iteration in order to ensure accuracy of the system, in particular with respect to energy conservation. Details of the method employed to couple the solvers can be found in Hall *et al.* [76].

3.4.1. Geometry Parameterisation

The parameterisation grid used in this problem is shown in figure 32 and consists of eight columns by four rows of cells. The baffle design is forced to by symmetric about the vertical centre line of the tank so the values in the parameterisation cells are mirrored across this line. The optimisation begins with all cells set to have a volume fraction of zero such that the initial geometry is a "clean" tank, and the total fluid volume is fixed.

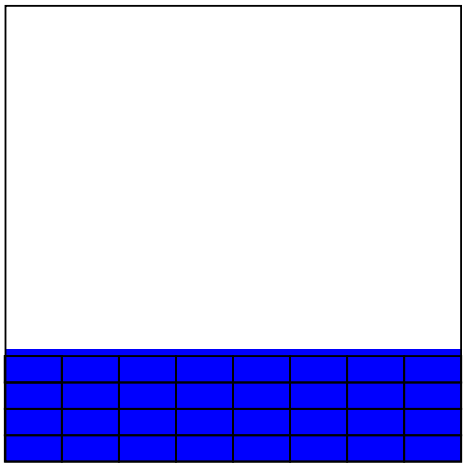
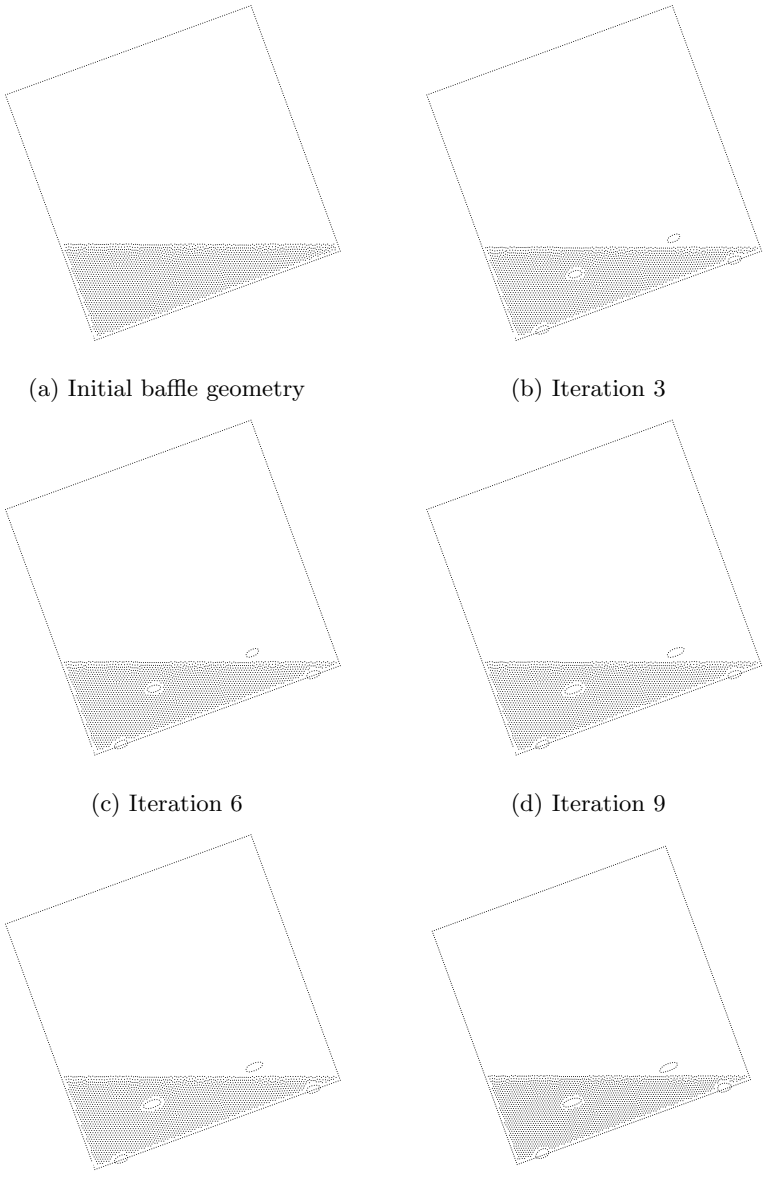


Figure 32: Diagram of baffle parameterisation grid overlaid on tank with settled fluid depth indicated by the blue shaded area

3.4.2. Baffle geometry

The geometry produced through the optimisation process is shown in figure 33f and consists of four separate baffle elements, two on the base of the tank and two near the free surface positioned roughly a third of the tank width from the walls and each other. The effect on the tank behaviour of the baffles on the base of the tank is difficult to discern, possibly they act to slightly raise the free surface level, as the volume of fluid in the tank is kept constant, or modify the wave speed just before the tank walls.

The effect of the baffle geometry can be seen in the time history of angular



(a) Initial baffle geometry

(b) Iteration 3

(c) Iteration 6

(d) Iteration 9

(e) Iteration 12

(f) Final baffle geometry

Figure 33: Baffle geometry optimisation iteration

position shown in figure 34 and in the plots of the fluid behaviour for the initial and optimised design in figure 35. The first negative peak shown in figure 34 is of a larger amplitude for the optimised case than for the clean tank. This is due to waves, travelling from right to left, encountering the first baffle and forcing the tank to travel further. However, after this the tank with baffles reverses direction faster as the fluid impact on the left hand wall is reduced and delayed, shown in figure 35, by the presence of the left hand baffle. The rightmost baffle then acts to reduce the amplitude of the wave to the right reducing the positive peak amplitude. Figure 36 shows the convergence history of the optimisation for this problem, demonstrating a reduction in the objective function of 2.2%, and again some non-monotonic convergence due to the projection process used during the line search in the conjugate gradient calculation is seen.

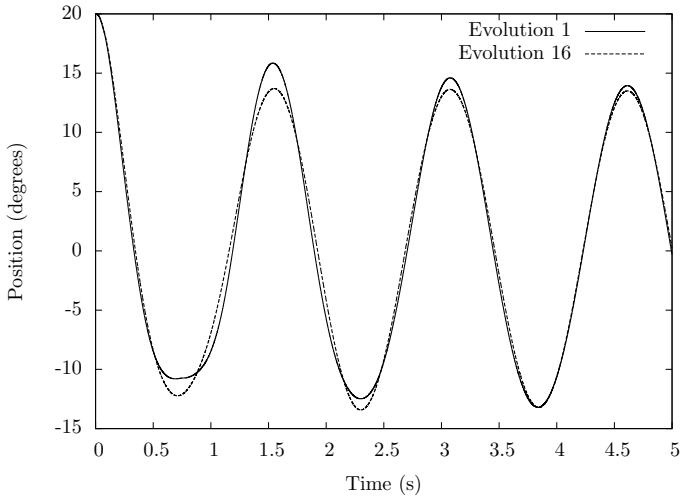
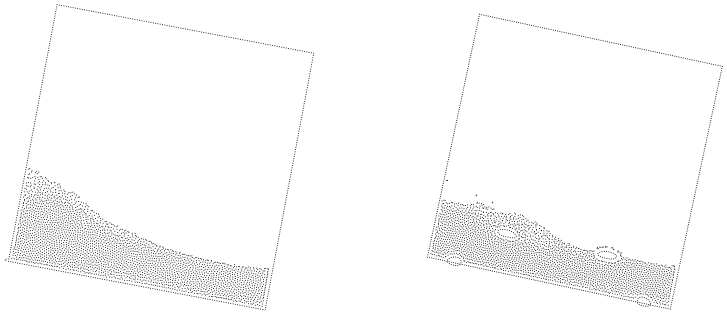


Figure 34: Time history of tank position for initial and final design iterations

4. Conclusions and Future Work

A new geometry and topology parameterisation scheme has been introduced which uses the volume fraction of solid within cells in a parameterisation grid



(a) Iteration 1 (b) Iteration 16

Figure 35: Fluid behaviour changes with iteration number at T=0.78s

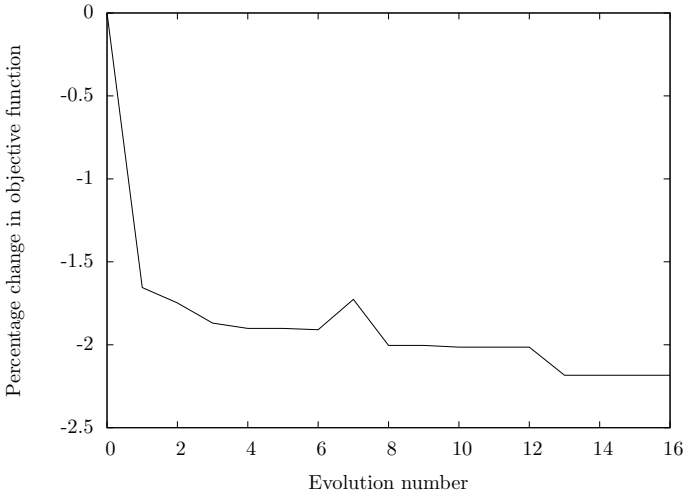


Figure 36: Convergence history of baffle optimisation problem

as design variables. This information can then be used to construct surfaces for use in optimisation problems. Results presented here have focussed first on examining the geometric abilities of the parameterisation through a qualitative test of the method's ability to reproduce a two-element aerofoil and a harbour outline, and subsequently on a quantitative analysis of the method's ability to reconstruct a set of common aerofoils.

The flexibility of the method is demonstrated by its ability to parameterise the outlines of the test images, which all have different topologies and surface characteristics, without changes to the method. It is also shown that the parameterisation scheme is capable of a high degree of accuracy as it successfully reconstructs a number of aerofoil geometries to within a tolerance typical of wind tunnel models. Although the parameterisation requires more design variables to achieve the same accuracy as other methods used for aerofoil parameterisation, those methods implicitly have information about the geometry of aerofoils encoded within them. As such, these methods perform better for aerofoil cases, but would be unlikely to perform so well for other problems, especially those involving topological change.

Three optimisation problems were considered: minimisation of drag on a geometry in supersonic flow, the design of a seawall geometry to prevent overtopping, and the design and placement of baffles to increase fluid damping in an oscillating tank. In all cases the initial design was improved. The Busemann drag minimisation problem demonstrated the benefit of having a parameterisation that is capable of multiple different topologies and which can work with a heuristic type optimiser. Additionally the use of the parameterisation to produce sharp, faceted geometries indicates the flexibility of the method. In contrast, the seawall optimisation uses the parameterisation to produce smooth surfaces, showing the range of surfaces that can be constructed. Finally, the

utility of having a parameterisation method capable of topological change was again demonstrated by showing how the parameterisation allows changing the number, placement and shape of baffles in a unified process for a sloshing tank. Future work will include additional improvements allowing refinement of parameterisation cells and extension to 3D.

Acknowledgements

The authors would like to acknowledge the funding for James Hall from the EPSRC DTP EP/K502996/1 at the University of Bristol. Data access: the data necessary to support the conclusions are included in the paper.

[1] B. Mohammadi and O. Pironneau. Shape optimization in fluid mechanics. *Annual Review of Fluid Mechanics*, 36:255 – 279, 2004.

[2] J. P. Reding and D. M. Jecmen. An advanced aerospike to minimise nose drag. *Lockheed Horizons*, 15:46 – 54, 1984.

[3] L.D. Huebner and A.M. Mitchell. Experimental results on the feasibility of an aerospike for hypersonic missiles. In *32nd Aerospace Sciences Meeting & Exhibit*. AIAA-95-0737, 1995.

[4] Matteo Diez, Emilio F. Campana, and Frederick Stern. Design space dimensionality reduction in shape optimization by Karhunen-Loève expansion. *Computer Methods in Applied Mechanics and Engineering*, 283:1525 – 1544, 2015.

[5] Matteo Diez, Andrea Serani, Emilio F. Campana, Silvia Volpi, and Frederick Stern. Design space dimensionality reduction for single- and multidisciplinary shape ptimization. In *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. AIAA-2016-4295, 2016.

[6] B. Raghavan, P. Breitkopf, Y. Tourbier, and P. Villon. Towards a space reduction approach for efficient structural shape optimization. *Structural and Multidisciplinary Optimization*, 48:987 – 1000, 2013.

[7] Brenda Kulfan and John Bussoletti. 'Fundamental' parameteric geometry representations for aircraft component shapes. In *Multidisciplinary Analysis Optimization Conferences*. AIAA-2016-4295, American Institute of Aeronautics and Astronautics, 2006.

[8] A. Sobester and S. Powell. Design space dimensionality reduction through physics-based geometry re-parameterization. *Optimisation and Engineering*, 14:37 – 59, 2013.

[9] Anthony Jameson and James Reuther. Control theory based airfoil design using Euler equations. In *Proceedings of 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City Beach*, pages 206 – 222, 1994.

[10] A. Jameson, S. Shankaran, and L.Martinelli. Continuous adjoint method for unstructured grids. *AIAA Journal*, 46(5):1226 – 1239, 2008.

[11] Rui Hu, Antony Jameson, and Qiqi Wang. Adjoint-based aerodynamic optimization of supersonic biplane airfoils. *Journal of Aircraft*, 49(3):802 – 814, 2012.

[12] A. Jameson, L. Martinelli, and N.A. Pierce. Optimum aerodynamic design using the NavierStokes equations. *Theoretical and Computational Fluid Dynamics*, 10(1-4):213 – 237, 1998.

[13] James J. Reuther, Antony Jameson, Juan J. Alonso, Mark J. Rimlinger, and David Saunders. Constrained multipoint aerodynamic shape optimiza-

tion using an adjoint formulation and parallel computers, part 1. *Journal of Aircraft*, 36(1):51–60, January 1999.

[14] Martin Burger, Benjamin Hackl, and Wolfgang Ring. Incorporating topological derivatives into level set methods. *Journal of Computational Physics*, 194(1):344 – 362, 2004.

[15] N.P. van Dijk, K. Maute, M. Lanelaar, and F. van Keulen. Level-set methods for structural topology optimization: a review. *Structural and Multidisciplinary Optimisation*, 34:437 – 472, 2013.

[16] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12 – 49, 1988.

[17] J.A. Sethian and Andreas Wiegmann. Structural boundary design via level set and immersed interface methods. *Journal of Computational Physics*, 163(2):489 – 528, 2000.

[18] Michael Yu Wang, Xiaoming Wang, and Dongming Guo. A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 192(12):227 – 246, 2003.

[19] Grégoire Allaire, François Jouve, and Anca-Maria Toader. Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics*, 194(1):363 – 393, 2004.

[20] Grégoire Allaire, François Jouve, and Anca-Maria Toader. A level-set method for shape optimization. *Comptes Rendus Mathematique*, 334(12):1125 – 1130, 2002.

[21] Peng Wei and Michael Yu Wang. Piecewise constant level set method

for structural topology optimization. *International Journal for Numerical Methods in Engineering*, 78(4):379 – 402, 2009.

[22] S. Kreissl, G. Pingen, and K. Maute. An explicit level set approach for generalised shape optimization of fluids with the lattice-boltzmann method. *International Journal for Numerical Methods in Fluids*, pages 496–519, 2011.

[23] S. Kreissl and K. Maute. Levelset based fluid topology optimization using the extended finite element method. *Structural and Multidisciplinary Optimisation*, 46:311 – 326, 2012.

[24] M. Hasan Imam. Three-dimensional shape optimization. *International Journal for Numerical Methods in Engineering*, 18(5):661 – 673, 1982.

[25] Wang Shu-Yu, Sun Yanbing, and R.H. Gallagher. Sensitivity analysis in shape optimization of continuum structures. *Computers & Structures*, 20(5):855 – 867, 1985.

[26] Ren-Jye Yang and Kyung K. Choi. Accuracy of finite element based shape design sensitivity analysis. *Journal of Structural Mechanics*, 13(2):223 – 239, 1985.

[27] V. Braibant and C. Fleury. Shape optimal design using B-splines. *Computer Methods in Applied Mechanics and Engineering*, 44(3):247 – 267, 1984.

[28] Thomas W. Sederberg and Scott R. Parry. Free-form deformation of solid geometric models. *SIGGRAPH Comput. Graph.*, 20(4):151 – 160, August 1986.

[29] Jamshid Samareh. Multidisciplinary aerodynamic-structural shape optimization using deformation (massoud). In *Multidisciplinary Analysis Optimization Conferences*. AIAA-2000-4911, American Institute of Aeronautics and Astronautics, 2000.

[30] Wataru Yamazaki, Sylvain Mouton, and Gerald Carrier. Geometry parameterization and computational mesh deformation by physics-based direct manipulation approaches. *AIAA Journal*, 48(8):1817 – 1832, August 2010.

[31] A. M. Morris, C.B. Allen, and T.C.S Rendall. CFD-based optimization of aerofoils using radial basis functions for domain element parameterization and mesh deformation. *International Journal for Numerical Methods in Fluids*, 58:827 – 860, 2008.

[32] Daniel J. Poole, Christian B. Allen, and T. C. S. Rendall. Application of control point-based aerodynamic shape optimization to two-dimensional drag minimization. In *AIAA SciTech*. AIAA-2014-0413, 2014.

[33] Kevin Lane and David Marshall. Inverse airfoil design utilizing CST parameterization. In *Aerospace Sciences Meetings*. AIAA-2010-1228, American Institute of Aeronautics and Astronautics, 2010.

[34] Jacob Haderlie and William Crossley. A parametric approach to supercritical airfoil design optimization. In *Aviation Technology, Integration, and Operations (ATIO) Conferences*. American Institute of Aeronautics and Astronautics, 2009.

[35] Helmut Sobieczky. Parametric airfoils and wings. In Kozo Fujii and George S. Dulikravich, editors, *Recent Development of Aerodynamic Design Methodologies*, volume 65 of *Notes on Numerical Fluid Mechanics (NNFM)*, pages 71–87. Vieweg+Teubner Verlag, 1999.

[36] Akira Oyama, Shigeru Obayashi, and Takashi Nakamura. Real-coded adaptive range genetic algorithm applied to transonic wing optimization. *Applied Soft Computing*, 1(3):179 – 187, 2001.

[37] Raymond M. Hicks and Preston A. Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15(7):407 – 412, July 1978.

[38] James J. Reuther, Antony Jameson, Juan J. Alonso, Mark J. Rimllnger, and David Saunders. Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 2. *Journal of Aircraft*, 36(1):61–74, January 1999.

[39] D. J. Poole, C. B. Allen, and T. C. S. Rendall. Metric-based mathematical derivation of efficient aerofoil design variables. *AIAA Journal*, 53(5):1349–1361, 2015.

[40] Daniel J. Poole, Christian B. Allen, and T. Rendall. Aerofoil design variable extraction for aerodynamic optimization. In *Fluid Dynamics and Co-located Conferences*. AIAA-2013-2705, American Institute of Aeronautics and Astronautics, 2013.

[41] Dominic A. Masters, Nigel J. Taylor, T. Rendall, Christian B. Allen, and Daniel J. Poole. A geometric comparison of aerofoil shape parameterisation methods. In *AIAA SciTech*. AIAA-2015-0761, American Institute of Aeronautics and Astronautics, 2015.

[42] C. W. Hirt and B.D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39:201–225, 1981.

[43] W.F. Noh and Paul Woodward. SLIC (simple line interface calculation). In AdriaanI. van de Vooren and PieterJ. Zandbergen, editors, *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28 July 2, 1976 Twente University, Enschede*, volume 59 of *Lecture Notes in Physics*, pages 330–340. Springer Berlin Heidelberg, 1976.

[44] B. Parker and D. Youngs. Two and three dimensional Eulerian simulation of fluid flow with material interfaces. Technical report, Atomic Weapons Establishment, 1992.

[45] Jie Li. Calcul d'interface affine par morceaux. *Comptes rendus de l'Académie des sciences. Série II*, 320(8):391 – 396, 1995.

[46] Thomas Borrvall and Joakim Petersson. Topology optimization of fluids in stokes flow. *International Journal for Numerical Methods in Fluids*, 41(1):77 – 107, 2003.

[47] G. Pingen, A. Evgrafov, and K. Maute. Topology optimsation of flow domains using the lattice Boltzmann method. *Structural and Multidisciplinary Optimisation*, 34:507 – 524, 2007.

[48] M.J. Krause, G. Th́ater, and V. Heuveline. Adjoint-based fluid flow control and optimisation with lattice boltzmann methods. *Computers and Mathematics with Applications*, 65:945–960, 2013.

[49] K. Yaji, T. Yamada, M. Yoshino, T. Matsumoto, K.Izui, and S. Nishiwaki. Topology optimization using the lattice boltzmann method incorporating level set boundary expressions. *Journal of Computational Physics*, 274:158 – 181, 2014.

[50] M.P. Bendsøe. Optimal shape design as a material distribution problem. *Structural Optimization*, 1(4):193– 202, 1989.

[51] M. P. Bendsøe and O. Sigmund. Material interpolation schemes in topology optimization. *Archive of Applied Mechanics*, 69(9-10):635 – 654, 1999.

[52] D. Isebe, P. Azerad, B. Mohammadi, and F. Bouchette. Optimal shape design of defense structures for minimizing short wave impact. *Coastal Engineering*, 55(1):35 – 46, 2008.

[53] I. Prilepov, H. Obermaier, E. Deines, C. Garth, and K.I. Joy. Cubic gradient-based material interfaces. *Visualization and Computer Graphics, IEEE Transactions on*, 19(10):1687 – 1699, Oct 2013.

[54] J.D. Deaton and R.V. Grandhi. A survey of structural and multidisciplinary continuum optimisation: post 2000. *Strutcural and multidisciplinary optimisaion*, 49:1 – 38, 2014.

[55] D.J. Munk, G.A. Vio, and G.P. Steven. Topology and shape optimization methods using evolutionary algorithms: a review. *Strutcural and multidisciplinary optimisaion*, 52:613 – 631, 2015.

[56] B. Bourdin and A. Chambolle. Design-dependent loads in topology optimization. *ESAIM: Control, Optimisation and Calculus of Variations*, 9:19 – 48, 2003.

[57] J.W. Miles and Y.K. Lee. Helmholtz resonance of harbours. *Journal of Fluid Mechanics*, 67(3):445 – 464, 1975.

[58] Gerald Farin. *Curves and Surfaces for CAGD A Practical Guide*. Academic Press Inc, London, 3 edition, 1993.

[59] Steven A. Coons. Surfaces for computer-aided design of space forms. Technical report, Massachusetts Institute of Technology, June 1967.

[60] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM National Conference*, ACM '68, pages 517–524, New York, NY, USA, 1968. ACM.

[61] W. Lorenson and H. Cline. Marching cubes: a high-resolution 3D surface construction algorithm. *Computer Graphics*, 21(4):163 – 168, 1987.

[62] S.S. Sarakinos, E. Amoiralis, and I.K. Nikolos. Exploring freeform deformation capabilities in aerodynamic shape parameterization. In *Computer as a*

Tool, 2005. EUROCON 2005.The International Conference on, volume 1, pages 535–538, Nov 2005.

[63] Patrice Castonguay and Siva Nadarajah. Effect of shape parameterization on aerodynamic shape optimization. In *Aerospace Sciences Meetings*. AIAA-2007-59, American Institute of Aeronautics and Astronautics, 2007.

[64] Arash Mousavi, Patrice Castonguay, and Siva Nadarajah. Survey of shape parameterization techniques and its effect on three-dimensional aerodynamic shape optimization. In *Fluid Dynamics and Co-located Conferences*. AIAA-2007-3837, American Institute of Aeronautics and Astronautics, 2007.

[65] A. Busemann. Aerodynamic lift at supersonic speeds. In 5^{th} *Volta Congress*, pages 315 – 347, 1935.

[66] Yuki Utsumi. Multidisciplinary design optimization of a three-dimensional supersonic biplane based on method of characteristics. In *27th International Congress of the Aeronautical Sciences*, 2010.

[67] Daniel J. Poole, Christian B. Allen, and Thomas Rendall. A constrained global optimization framework. In *AIAA Aviation*. AIAA-2014-2034, American Institute of Aeronautics and Astronautics, June 2014.

[68] D.J. Poole, C.B. Allen, and T.C. S. Rendall. A generic framework for handling constraints with agent-based optimization algorithms and application to aerodynamic design. *Optimization and Engineering*, 2016. In press.

[69] A.D. Belegundu and J.S. Aurora. A study of mathematical programming methods for structural optimization. part ii: numerical results. *International Journal for Numerical Methods in Fluids*, 21(9):1601–1623, 1985.

[70] B.K. Kannan and S.N. Kramer. An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Journal of Mechanicsl Design*, 116(2):405–411, 1994.

[71] S.S. Rao. *Engineering optimization: theory and practice*. Wiley, 2009.

[72] A. Crespo, M. Gomez-Gesteria, and R. Dalyrmple. Modeling dam break behavior over a wet bed by a SPH technique. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 134(6):313–320, 2008.

[73] J. J. Monaghan. Simulating free surface flows with SPH. *Journal of Computational Physics*, 110:399 – 406, 1994.

[74] J. B. Rosen. The gradient projection method for nonlinear programming. part I. linear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 8(1):181–217, 1960.

[75] J. G. Anderson, S. E. Semercigil, and Ö. F. Turan. A standing-wave-type sloshing absorber to control transient oscillations. *Journal of Sound and Vibration*, 232(5):839 – 856, 2000.

[76] J. Hall, T.C.S. Rendall, C.B. Allen, and H. Peel. A multi-physics computational model of fuel sloshing effects on aeroelastic behaviour. *Journal of Fluids and Structures*, 56:11 – 32, 2015.